

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-068075

(43)Date of publication of application : 11.03.1994

(51)Int.Cl.

G06F 15/20
G06F 15/20
G06F 3/153
G09G 5/22

(21)Application number : 04-217827

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 17.08.1992

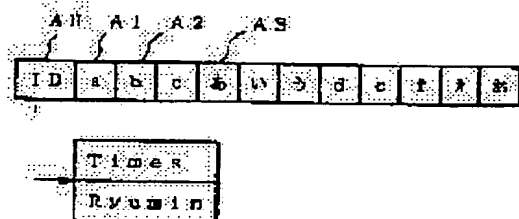
(72)Inventor : OGIKUBO TOMOFUMI

(54) FONT DATA STORAGE SYSTEM

(57)Abstract:

PURPOSE: To efficiently perform even copy processing and paste processing by efficiently storing data in a short time.

CONSTITUTION: In a system for storing the character data of character strings whose font formats are different in a text buffer, the text buffer is provided with character code areas A1, A2,... for storing the respective character codes of the character strings and a font format area AH for storing all the font format codes of the character strings altogether provided at the head of the areas.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁(J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-68075

(43)公開日 平成6年(1994)3月11日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/20	5 3 0 E	9288-5L		
	5 6 2 N	9288-5L		
3/153	3 1 0 B	7165-5B		
G 0 9 G 5/22		9061-5G		

審査請求 未請求 請求項の数1(全 14 頁)

(21)出願番号 特願平4-217827

(22)出願日 平成4年(1992)8月17日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 荻久保 友史

東京都府中市東芝町1番地 株式会社東芝
府中工場内

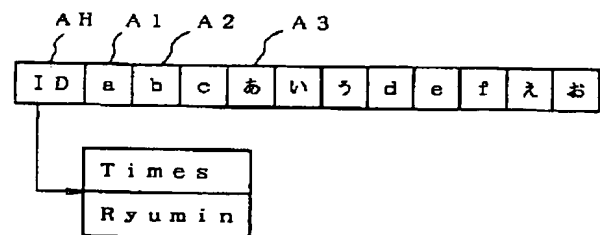
(74)代理人 弁理士 三好 秀和 (外1名)

(54)【発明の名称】 フォントデータ格納方式

(57)【要約】

【目的】 短時間で効率の良いデータ格納処理を可能とし、これによりコピー処理やペースト処理をも効率的に行うことを可能とする。

【構成】 フォント書式の異なる文字列の文字データをテキストバッファに格納する方式において、前記テキストバッファは、前記文字列の各文字コードを格納する文字コードエリアA1、A2、…と、このエリアの先頭に設けられ前記文字列の全てのフォント書式コードを一括して格納するフォント書式エリアAHとを有する。



【特許請求の範囲】

【請求項1】 フォント書式の異なる文字列の文字データをテキストバッファに格納する方式において、前記テキストバッファは、前記文字列の各文字コードを格納する文字コードエリアと、このエリアの先頭に設けられ前記文字列の全てのフォント書式コードを一括して格納するフォント書式エリアとを有することを特徴とするフォントデータ格納方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、DTP (Desk Top Publishing)等においてフォント書式の異なる文字列を表示する場合に使用されるテキストバッファに対するフォントデータ格納方式に関する。

【0002】

【従来の技術】 DTP等においては、フォント書式の異なる種類のフォントを用い、ひらがな文字や英文字等が混在した文書を作成し、画面表示またはプリントアウトする人が多い。従来、例えば、図10に示すように、フォント書式の異なる英数字とひらがな文字から成る文字列“a b c あいう d e f えお”を画面表示またはプリントアウトさせるためには、以下のようにしてテキストバッファにデータを格納していた。

【0003】 今、英数字のフォント名が“Times (タイムズ)”であり、ひらがな文字のフォント名が“Ryumin (リュミン; 明朝体)”であるとする。この場合、テキストバッファには、図11に示すように、フォント名の変わり目のエリアにフォント名を示すフォントIDを格納しておき、このフォント名の後に、そのフォント名の文字コードを格納する。図11においては、先頭エリアに英数字“a b c”のフォント名“Times”を示すコードを格納し、その次のエリアから“a b c”各英数字のコードを格納する。次にひらがな文字“あいう”のフォント名“Ryumin”を示すコードを格納し、その次のエリアからひらがな文字“あいう”の各ひらがな文字コードを格納する。さらに英数字“d e f”のフォント名コード“Times”を格納し、その次に各英数字コードを格納する。このように、フォント名が変わる都度、各文字コード列の先頭エリアにフォント名を格納するようしていた。

【0004】 その手順を図12のフローチャートに基づいて更に詳説する。

【0005】 初めにフラグ1およびフラグ2を“0”にするとともに、テキストバッファをクリアする(ステップST101)。

【0006】 次にテキストバッファの先頭エリアに英数字フォントIDを挿入する(ステップST102)。次にオペレータにより文字が入力されるとその文字が英数字か否かが判定される(ステップST103, ST104)。若し、入力文字が英数字であれば、フラグ2を

“0”にセット(ステップST105)し、フラグ1=フラグ2を確認する。最初は、フラグ1=フラグ2=“0”が成立する(ステップST106Y)ので、テキストバッファに文字を追加していく(ステップST107)。そして、フラグ2の内容(この場合は“0”)をフラグ1にセットした後、ステップST103に戻り、再び文字入力を実行する(ステップST108)。

【0007】 英数字入力中に英数字以外の文字が入力された場合(ステップST104N)には、フラグ2を“1”にセット(ステップST109)し、フラグ1=フラグ2を確認する(ステップST106)。この場合には、フラグ1≠フラグ2であり、かつフラグ2=“1”である(ステップST106N, ST110N)から、テキストバッファに漢字用フォントIDを追加する(ステップST111)。その後、テキストバッファに入力された漢字またはひらがな文字を追加し(ステップST107)、フラグ2の内容(=“1”)をフラグ1にセットした後、再びステップST103に戻る(ステップST108)。

【0008】 さらに次に入力された文字が英数字に変わっている場合(ステップST104Y)には、フラグ2(=“1”)を“0”にセット(ステップST105)し、フラグ1=フラグ2を確認する。このときには、フラグ1≠フラグ2であり、かつフラグ2=“0”である(ステップST106N, ST110Y)から、テキストバッファに英数字用フォントIDを追加する(ステップST112)。その後、テキストバッファに入力された英数字を追加し(ステップST107)、フラグ2の内容(=“1”)をフラグ1にセットした後、再びステップST103に戻る(ステップST108)。

【0009】 このように複雑な処理を実行した後、テキストバッファにフォントIDが格納されるのである。

【0010】 図13には、フォントIDが“Ryumin”と“Times”から成る文字列「a b c あいう d e f えお g h i かきく j k l け m n o さしす o p r セ」を格納するファイル(MIFファイル)形式の従来例を示す。図から理解されるように、フォントIDが変更される都度、フォントIDを付加するので、その手順は複雑なものとなっている。

【0011】 次に、表示された文字列の中でマウス等で選択された部分のフォント書式をコピーして異なるフォント書式(例えば、Helvetica, Gothic)から成る他の文字列へペーストする処理の従来例について説明する。

【0012】 例えば、図10において、範囲Aの部分「e f え」をコピーする場合には、英字の“e f”は“Times”であり、ひらがなの“え”は“Ryumin”であり、英数字フォントTimesと日本語フォントRyuminをそれぞれ取り出さなければならぬ。図14は従来のフォントコピー処理の手順を示し、

図15は従来のフォント書式ペースト処理の手順を示している。

【0013】図14において、先ず文字列の中からコピーしたい部分を選択してテキストバッファ内へ取り込む(ステップST121)。図10の例では、「efえ」が取り込まれる。次に、バッファの中からフォントIDが検索、抽出される(ステップST122)。図10の例では、“Times, Ryumin”が検索され抽出される。この場合、英字の“ef”は“Times”であり、ひらがなの“え”は“Ryumin”であり、英字フォントID“Times”と日本語フォントID“Ryumin”をそれぞれ取り出さなければならない。こうして、英数字フォントIDおよび漢字フォントIDが全て揃うと(ステップST123Y)、抽出したフォントIDをテキストバッファのクリップボードに格納する(ステップST124)。若し、選択部分のIDが見付からない場合には、テキストバッファを遡ってフォントIDを抽出する(ステップST125)。

【0014】このようにしてコピーされたフォント情報を他の文字列にペーストする場合は、図15において、先ず、ペースト先の文字列の中からマウス等で選択された部分をバッファに取り込む(ステップST131)。次に、この選択部分の英数字フォントIDおよび漢字フォントIDを全て検索し抽出する(ステップST132)。次いで、バッファの先頭が英数字の場合(ステップST133Y)には、英数字フォントIDをバッファの先頭に挿入する(ステップST134)。バッファの先頭が英数字でない場合(ステップST133N)には、漢字フォントIDをバッファの先頭に挿入する(ステップST135)。

【0015】次に、バッファを走査しフォントIDを抽出する(ステップST136)。抽出されたフォントIDが英数字フォントIDであれば(ステップST137Y)、先にクリップボード内に保存されている英数字フォントIDと交換する(ステップST138)。抽出されたフォントIDが英数字フォントIDでなければ(ステップST137N)、先にクリップボード内に保存されている漢字フォントIDと交換する(ステップST139)。こうしてバッファの走査が終了すると(ステップST140Y)、ステップST132の処理で抽出されたフォントIDをバッファの最後尾に追加する(ステップST141)。最後にこのようにして編集された選択部分を当初のバッファの対応部分と入れ替える(ステップST142)。これにより、選択部分が新たなフォントIDを含むデータとなり、ペースト処理が終了する。

【0016】

【発明が解決しようとする課題】このように、従来のコード格納方式によれば、図12のフローチャートおよび図13に示したような複雑の入力処理手順を踏んでお

り、頻繁に英字・漢字が交互に現れてくるような文字列の場合には、多くのフォント情報が必要となり、データ格納処理に長い時間がかかり処理効率が悪いという問題がある。また、図14および図15のフローチャートに示したようなコピー処理およびペースト処理も複雑となり、効率的でない。

【0017】本発明は上記の事情に鑑みてなされたものであり、その目的は、短時間で効率の良いデータ格納処理を可能とし、これによりコピー処理やペースト処理をも効率的に行うことができるフォントデータの格納方式を提供することにある。

【0018】

【課題を解決するための手段】上記の目的を達成するために本発明は、フォント書式の異なる文字列の文字データをテキストバッファに格納する方式において、前記テキストバッファは、前記文字列の各文字コードを格納する文字コードエリアと、このエリアの先頭に設けられ前記文字列の全てのフォント書式コードを一括して格納するフォント書式エリアとを有することを特徴とする。

【0019】

【作用】上記構成によれば、フォント書式コード(フォントID)が全てテキストバッファの先頭エリアに格納されているので、フォント書式が異なる毎にフォントIDを挿入している従来に比較して、格納処理手順が簡単になるとともに、コピー処理やペースト処理を効率良く行うことができる。

【0020】

【実施例】図1は本発明方式が適用されたテキストバッファの構成を示し、図1は本発明方式が適用されたパーソナルコンピュータの全体構成を示している。

【0021】初めに、このパーソナルコンピュータの全体構成を説明すると、キーボードやマウス等から成る入力部1と、入力されたデータを編集する編集処理部2と、編集処理部2で編集されたデータ等を保存する保存部3と、表示用データを作成する表示処理部4と、CRTを含み、作成された表示用データを画面表示するための画面処理部5と、プリンタを含み、作成された表示用データから印刷用データを生成してプリントアウトする印刷処理部6とから構成されている。前記編集処理部2は、さらに、文書編集部21と図形処理部22とから成り、また表示処理部4はテキスト表示処理部41と図形表示処理部42とから成っている。

【0022】図1に示すテキストバッファは、前記文書編集部21内に設けられ、先頭エリア(フォント書式エリア)AHにフォント情報のポインタID(フォントID)を一括格納し、その後続くエリアA1, A2, A3, …に表示文字のコードを格納している。すなわち、先頭エリアAHには、その後続く表示文字列で使用されるフォント情報のIDが全て格納されている。図の例では、TimesとRyuminを格納している。

【0023】図3は、このテキストバッファにフォントIDを格納するための入力処理の手順を示している。

【0024】先ずテキストバッファがクリアされ（ステップST1）した後、テキストバッファの先頭エリアAHにフォントIDが挿入される（ステップST2）。このIDは、表示される文字列の全てのフォントIDを含んでいる。

【0025】次にテキストバッファのエリアA1, A2, A3, …に順番に文字列のコードを入力していけば良い（ステップST3～ST4）。

【0026】図4は、前記従来例の図13に対応するMIFファイル形式を示しており、両図を比較すると良く分かるように、本実施例ではその処理手順が極めて簡素化され、そのデータ量が大幅に縮小される。

【0027】次にこのように格納されたテキストバッファの情報からその文字列を表示する処理を図5のフローチャートに基づいて説明する。この処理は表示処理部4のテキスト表示処理部41で実行される。

【0028】初めに、フラグ1およびフラグ2をとともに“0”クリアし（ステップST11）、仮に、英数字フォントを設定する（ステップST12）。次にテキストバッファから1文字を読み込み、これが英数字であれば（ステップST14Y）、フラグ2＝“0”を確認し（ステップST15）、次いで、フラグ1＝フラグ2を条件として文字を表示する（ステップST17）。その後、ステップST3に戻り、次に1文字をテキストバッファから読み込む。このとき、読み込まれた文字が英数字でなければ（ステップST14N）、フラグ2を

“1”にセットする（ステップST19）。次に、フラグ1＝フラグ2を確認するが、このときには、フラグ1≠フラグ2であり、かつフラグ2≠“1”であるので、漢字フォントが設定される（ステップST16N, ST20N, ST21）。そして、フラグ1の内容をフラグ2にセットした後（ステップST22）、文字表示がされる（ステップST17）。その後、ステップST3に戻り、次に1文字をテキストバッファから読み込む。このとき、読み込まれた文字が英数字であれば、フラグ2を“0”にする（ステップST15）。次に、フラグ1＝フラグ2であるか否かを判定するが、このとき、前回表示処理でフラグ1が“1”となっているので、ステップST20に進む。ここでフラグ2＝0が判別されるが、このときにはフラグ2＝0となっているので、英数字フォントが設定される（ステップST23）。そして、フラグ1の内容をフラグ2にセットした後（ステップST22）、文字表示がされる（ステップST17）。その後、ステップST3に戻り、次に1文字をテキストバッファから読み込む。このようにして、フォント情報に対応した文字列が表示されるのである。

【0029】次に、このようにして表示された文字列からフォントをコピーして他の文字列へペーストする処理

について説明する。

【0030】例えば、図6に示すように、文字列「a b c d e f g あいうえおかき」のフォントが“Ryumin”と“Times”で構成され、文字列「h i j k l m n さしすせそたち」のフォントが“Gothic”と“Helvetica”で表示されている場合に、図中斜線で囲む選択部分（表示画面中では反転部分）「f g あい」のフォント書式をコピーして、「mn さしす」へペーストする処理を例に説明する。

10 【0031】図7において、先ず、選択部分「f g あい」をテキストバッファに取り込み、テキストバッファの中からのフォントIDを検索して抽出する（ステップST31, ST32, ST33）。このとき抽出されたフォントIDは“Times, Ryumin”である。次に、抽出されたフォントIDをクリップボードに格納する（ステップST34）。なお、若しフォントIDが見つからない場合には、選択部分を遡ってフォントIDを抽出する（ステップST35）。

20 【0032】次に図8のペースト処理を実行すると、ペースト先の選択部分「mn さしす」をバッファに取り込み、そのテキストバッファのフォントIDを検索、抽出する（ステップST41, 42）。このとき抽出されたフォントIDは“Gothic, Helvetica”である。次に前記クリップボードのフォントID（Ryumin; Times）をテキストバッファの先頭に挿入するとともに、抽出されたフォントID（Gothic, Helvetica）をテキストバッファの最後尾に追加する（ステップST43, ST44）。そして、前記選択部分を先に格納されているテキストバッファの部分と入れ替える（ステップST45）。そのときのテキストバッファ内の構成を図9に示す。このように、フォント書式の複写処理を極めて容易に行うことができる。

30 【0033】次に本発明のカatalog機能について説明する。

【0034】このカatalog機能は、あるフォント書式に対して名前を付けておき、テキストに対してその名前を適用することで、簡単にフォントの変更ができるようにしたものである。

40 【0035】例えば、フォント書式がGothic 10pt（10ポイントのゴシック文字）と、Helvetica 12 pt（12ポイントのヘルベチカ文字）というフォント書式について、例えば“Body”という名前で予め登録しておき、Ryumin 10pt（10ポイントの明朝文字）TとTimes 12pt（12ポイントのタイムズ文字）から成る文字列の選択部分にフォント書式Gothic 10pt, Helvetica 12 ptをペーストするには、フォント書式“Body”を指定して、文字列の選択部分にペーストすれば良い。本発明の格納方式であれば、このような

カタログ機能を極めて容易に行うことができる。

【0036】

【発明の効果】以上説明したように本発明によれば、短時間で効率の良いデータ格納処理を可能とし、これによりコピー処理やペースト処理をも効率的に行うことができる。

【図面の簡単な説明】

【図1】本発明が適用されたテキストバッファの構成を示す説明図である。

【図2】本発明が適用されたパーソナルコンピュータの全体構成を示す説明図である。

【図3】テキストバッファへの入力格納処理の手順を示すフローチャートである。

【図4】本発明が適用されたMIFファイル形式を示す説明図である。

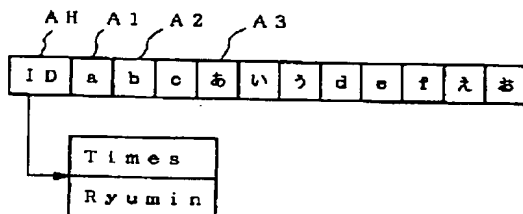
【図5】本発明における画面表示処理の手順を示すフローチャートである。

【図6】フォント書式のコピー処理およびペースト処理を示す説明図である。

【図7】本発明におけるフォント書式のコピー処理の手順を示すフローチャートである。

【図8】本発明におけるフォント書式のペースト処理の手順を示すフローチャートである。

【図1】



【図9】本発明におけるフォント書式のペースト処理実行後のテキストバッファの構成を示す説明図である。

【図10】画面上に表示されたフォント書式の異なる文字列を示す説明図である。

【図11】従来のテキストバッファの構成を示す説明図である。

【図12】従来におけるフォント書式入力処理の手順を示すフローチャートである。

【図13】従来におけるMIFファイル形式を示す説明図である。

【図14】従来におけるフォント書式コピー処理の手順を示すフローチャートである。

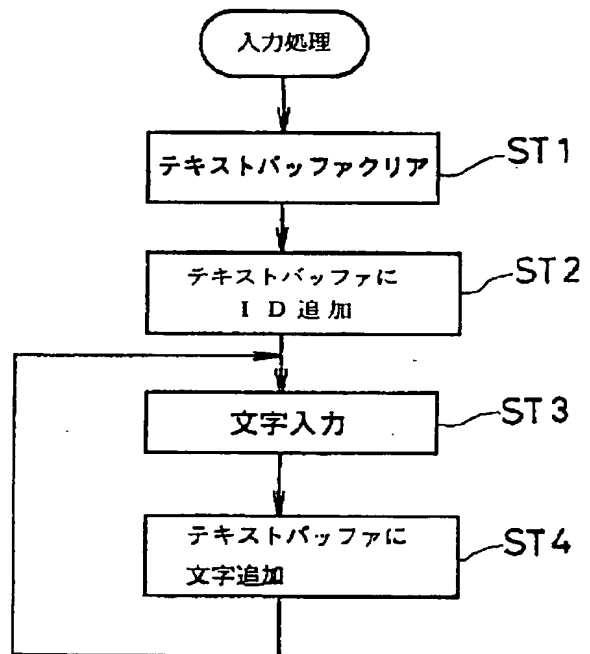
【図15】従来におけるフォント書式ペースト処理の手順を示すフローチャートである。

【符号の説明】

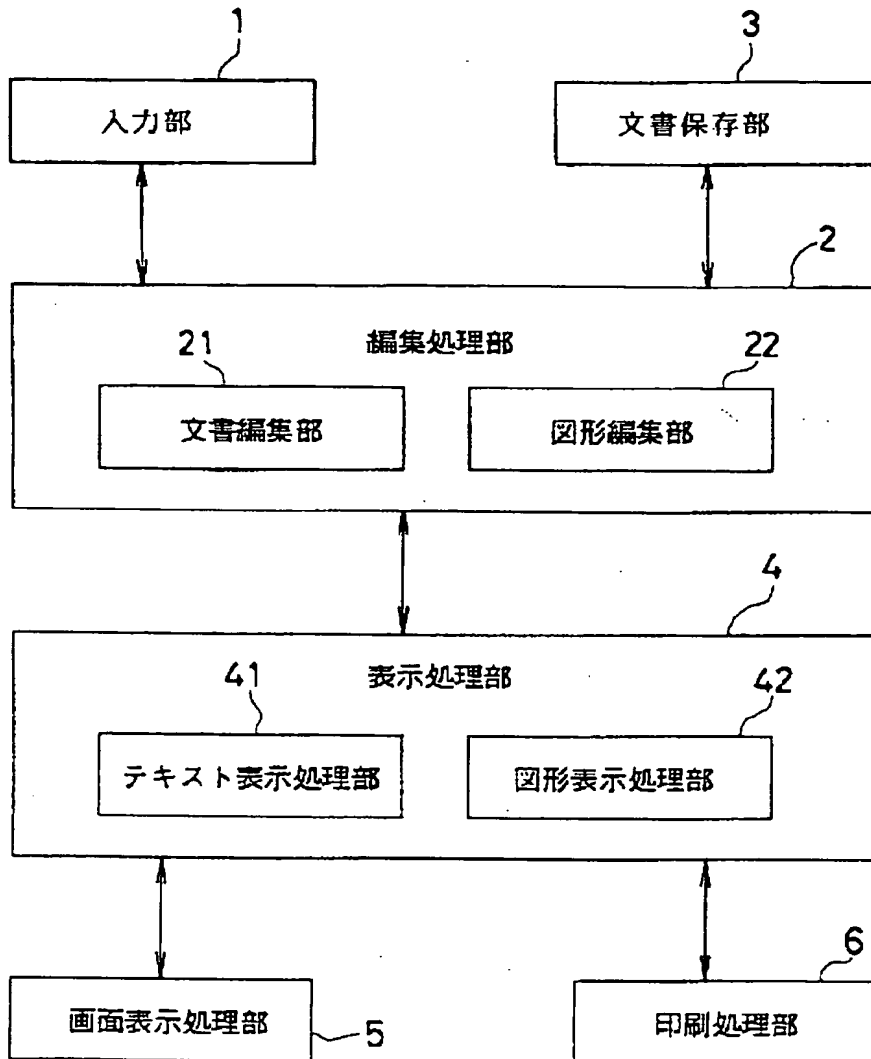
- 1 入力部
- 2 編集処理部
- 3 文書保存部
- 4 表示処理部
- 5 画面表示処理部
- 6 印刷処理部

AH 先頭エリア（フォント書式エリア）

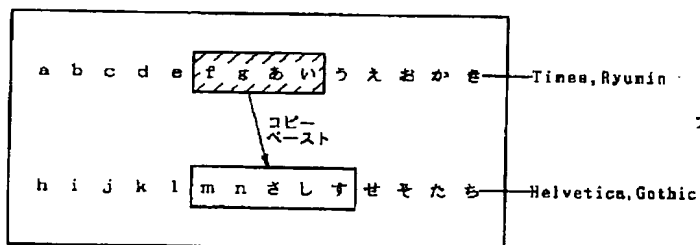
【図3】



【図2】



【図6】



【図10】



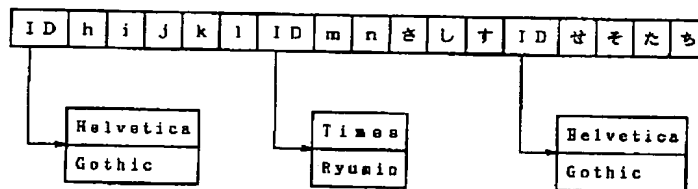
【図4】

```

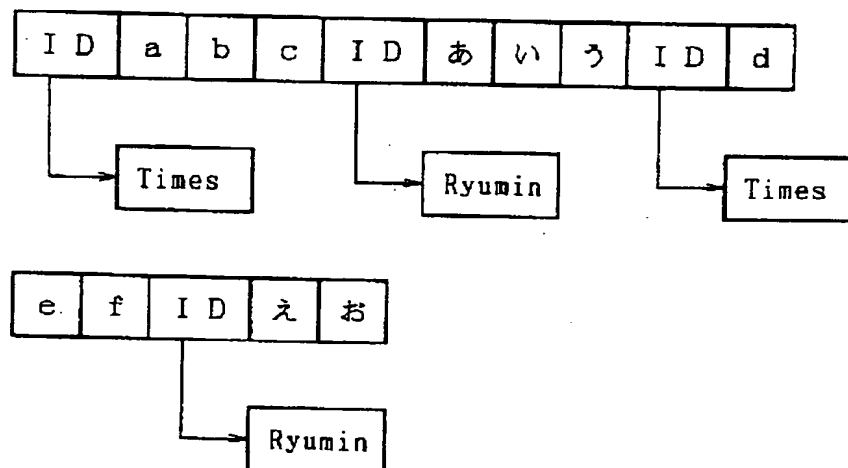
<Para
<ParaLine
<String 'abcあいうdefえおghiかきくjklけこmnoさしすoprせそ'>
>
> # end of Para

```

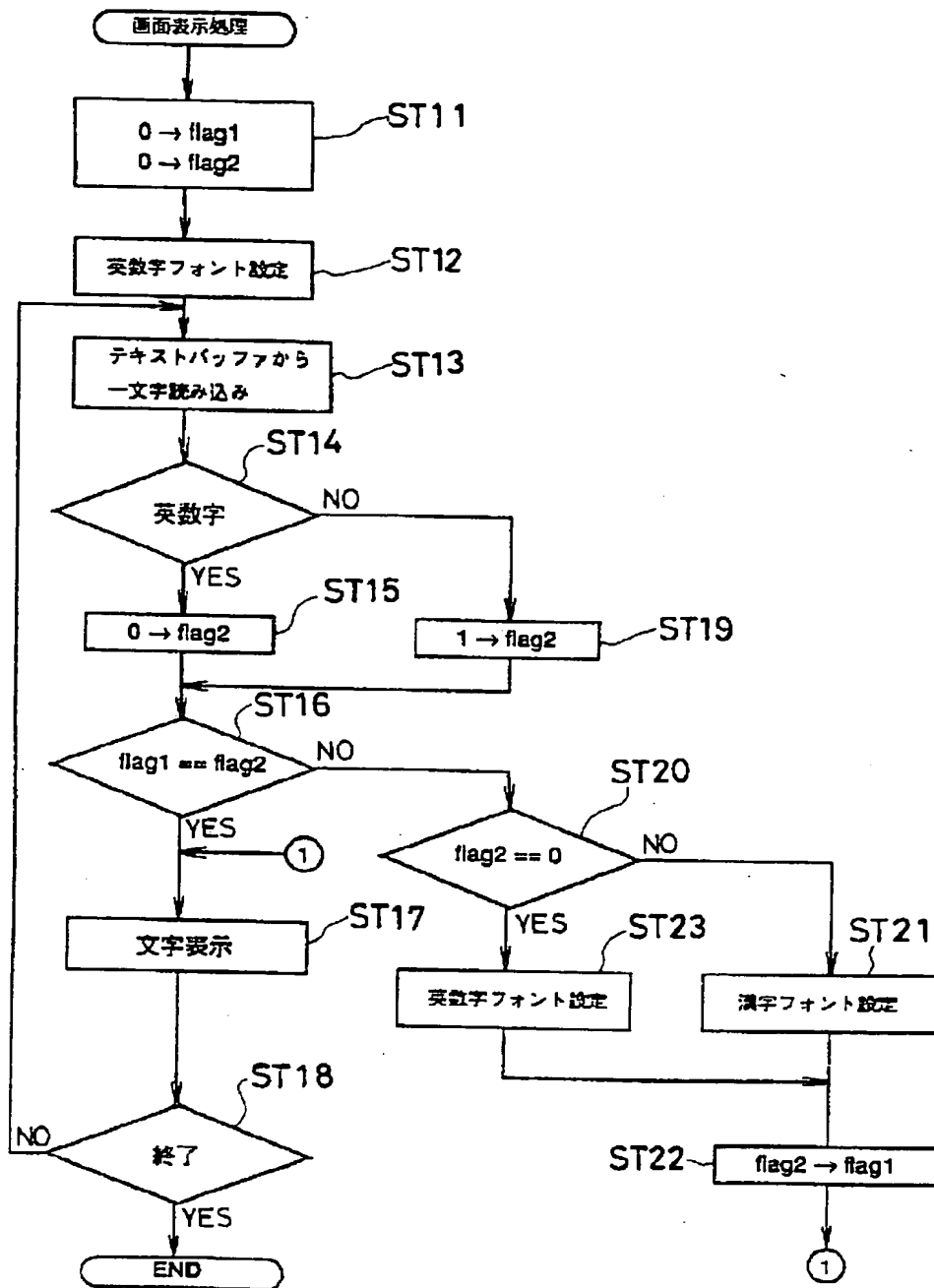
【図9】



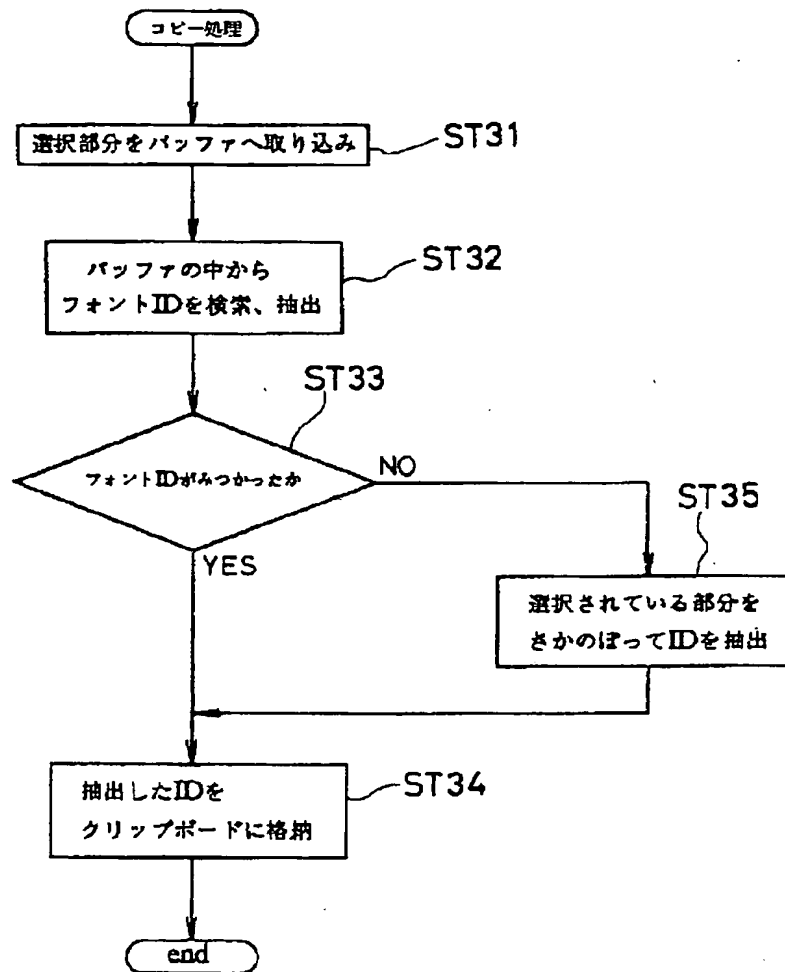
【図11】



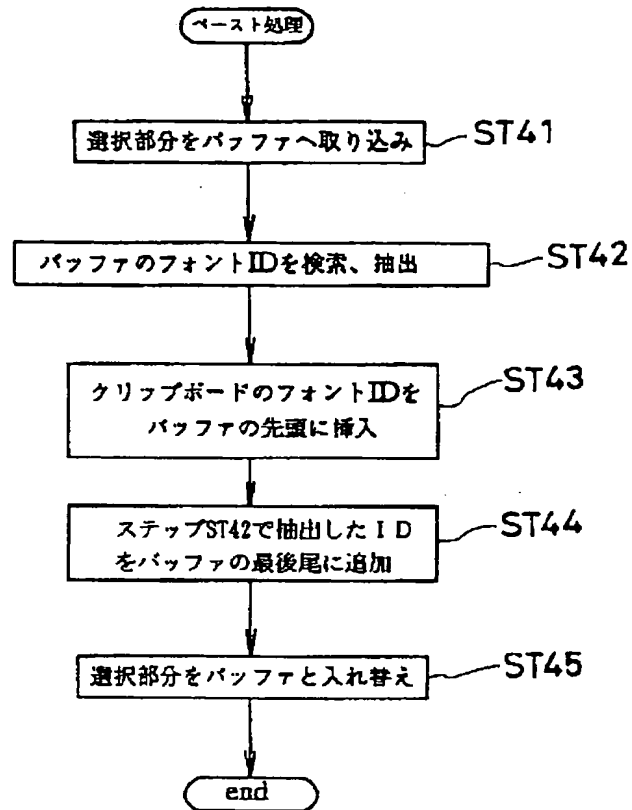
【図5】



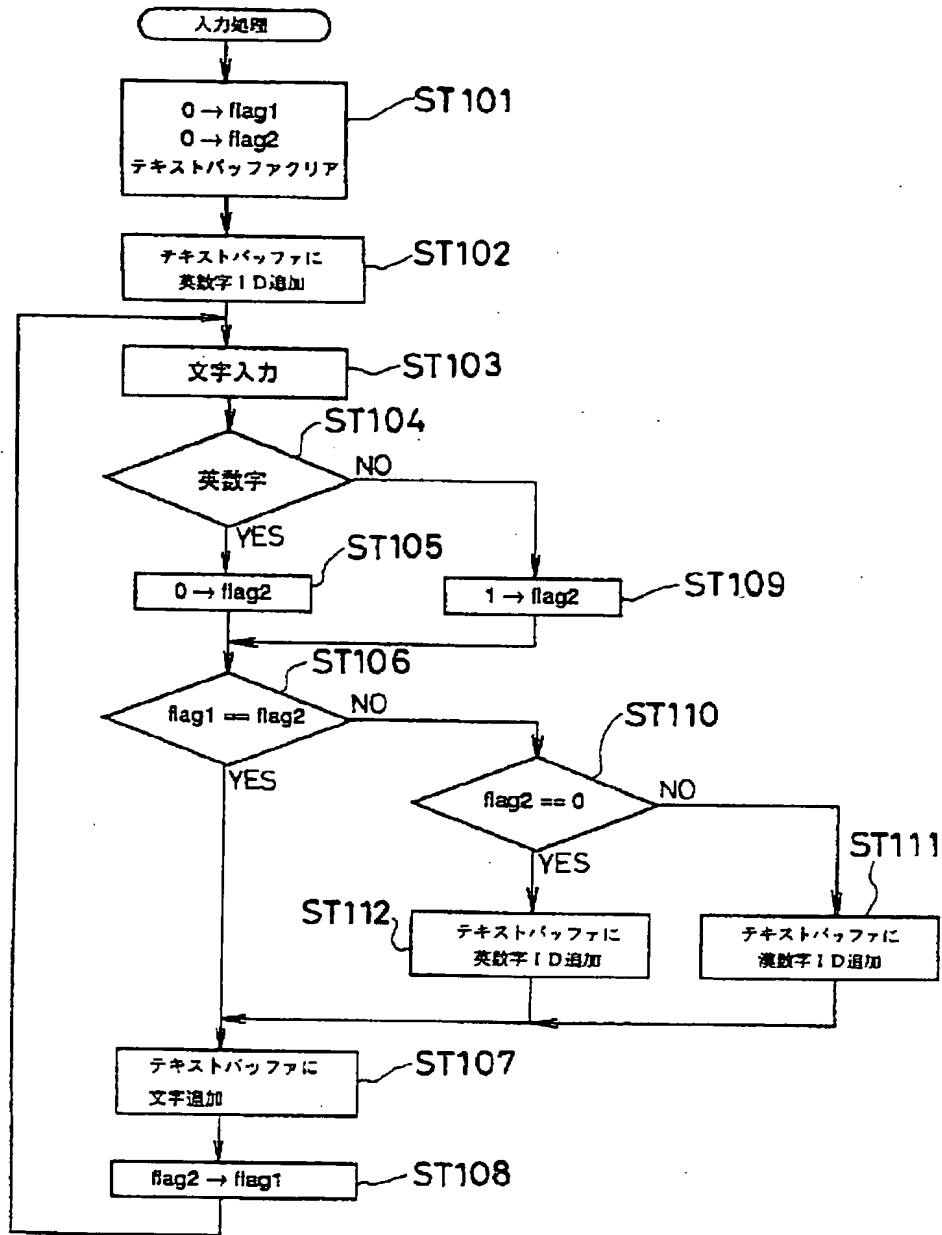
【図7】



【図8】



【図12】



【図13】

```

<ParaLine
<String 'abc'>
  <Font
    <FFamily '明朝'>
    <FSize 10>
  > # end of Font
  <String 'あいう'>
  <Font
    <FFamily 'Times'>
    <FSize 12>
  > # end of Font
  <String 'def'>
  <Font
    <FFamily '明朝'>
    <FSize 10>
  > # end of Font
  <String 'えお'>
  <Font
    <FFamily 'Times'>
    <FSize 12>
  > # end of Font
  <String 'ghi'>
  <Font
    <FFamily '明朝'>
    <FSize 10>
  > # end of Font
  <String 'かきく'>
  <Font
    <FFamily 'Times'>
    <FSize 12>
  > # end of Font
  <String 'jkl'>

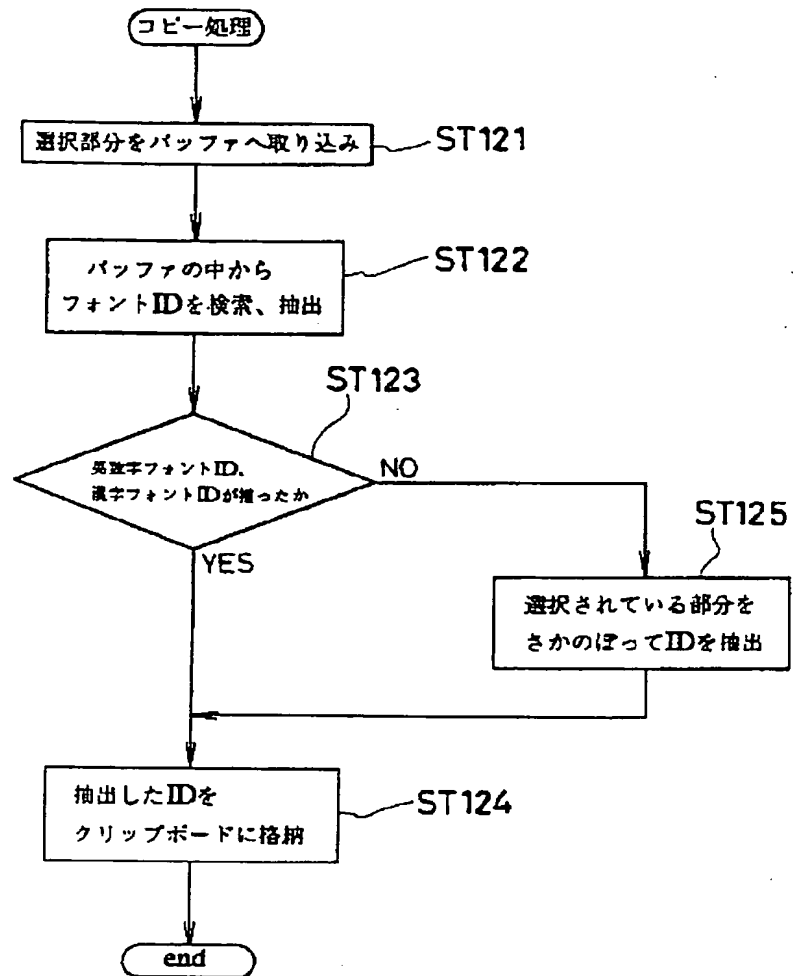
```

```

<Font
  <FFamily '明朝'>
  <FSize 10>
> # end of Font
<String 'けこ'>
<Font
  <FFamily 'Times'>
  <FSize 12>
> # end of Font
<String 'mno'>
<Font
  <FFamily '明朝'>
  <FSize 10>
> # end of Font
<String 'さしす'>
<Font
  <FFamily 'Times'>
  <FSize 12>
> # end of Font
<String 'opr'>
<Font
  <FFamily '明朝'>
  <FSize 10>
> # end of Font
<String 'せそ'>
<Font
  <FFamily 'Times'>
  <FSize 12>
> # end of Font
>
> # end of Para

```

【図14】



【図15】

